A new stochastic optimization algorithm to decompose large nonnegative tensors

Xuan Thanh VU[†§], Sylvain MAIRE[†], Caroline CHAUX[§], and Nadège THIRION-MOREAU[*†]

[†]Aix-Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, F-13397 Marseille, France

Université de Toulon, CNRS, LSIS, UMR 7296, F-83957 La Garde, France

[§]CNRS, Centrale Marseille, I2M, UMR 7373, F-13453 Marseille, France

{maire, thirion}@univ-tln.fr vuthanh.xuan.k4@gmail.com caroline.chaux@univ-amu.fr

Abstract

In this letter, the problem of nonnegative tensor decompositions is addressed. Classically, this problem is carried out using iterative (either alternating or global) deterministic optimization algorithms. Here, a rather different stochastic approach is suggested. In addition, the ever-increasing volume of data requires the development of new and more efficient approaches to be able to process "Big data" tensors to extract relevant information. The stochastic algorithm outlined here comes within this framework. Both flexible and easy to implement, it is designed to solve the problem of the CP (Candecomp/Parafac) decomposition of huge nonnegative 3-way tensors while simultaneously enabling to handle possible missing data.

Index Terms

Nonnegative Tensor Factorization (NTF); multi-linear algebra; Candecomp/Parafac (CP) decomposition; stochastic optimization; Big data/tensors; missing data

I. Introduction

The problem of tensor decompositions has gained a growing attention from different scientific communities due to its usefulness in various application fields (statistics, psychometrics, neurosciences,

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Manuscript submitted on February, 2015

This work has been carried out in the framework of the Labex Archimède (ANR-11-LABX-0033).

* Nadège THIRION-MOREAU is the corresponding author (email: thirion@univ-tln.fr).

chemometrics, numerical linear algebra, computer vision, linguistics, numerical analysis, data mining, biomedical engineering, (audio) signal processing, telecommunications and so on), see for example [1][2][3][4] for an overview. It has given rise to many works over the recent years. In this letter, we focus on one particular tensor decomposition known as the "Canonical Polyadic" decomposition. It consists of decomposing a tensor into a sum of rank-1 tensors. It can be either seen as a generalization of the matrix Singular Value Decomposition (SVD) to tensors or a special case of another tensor decomposition known as the Tucker decomposition [5] where the core tensor is restricted to be "diagonal". Depending on the considered community, different names have been used: Canonical Polyadic, Candecomp, CanD, Parafac (for PARAllel FACtor analysis), yet, the most popular acronym remains CP.

Most algorithms suggested to tackle the CP decomposition problem rely on the use of a well-chosen objective function and an iterative (either alternating or global) deterministic optimization algorithm. Direct solutions have been suggested too *e.g.* the GRAM-DTLD method [6][7]. Here, instead, we suggest a different stochastic optimization approach where random iterates are used. This genetic like algorithm might be considered as a special case of memetic algorithms [8][9]. In the case of CP decompositions, we will emphasize all the advantages one can find in restricting this population-based search to the case of two agents in the considered population. This will bring us to clearly delineate the most important milestones of the suggested approach.

Moreover, in a number of leading application areas of tensors (like fluorescence spectroscopy [10][11] or image processing (remote sensing and hyperspectral imaging [12]) for example) the data sought (*i.e.* the constituent vectors of the loading matrices involved in the CP decomposition) should be nonnegative since they stand for intrinsically nonnegative physical quantities (for example emission and excitation spectra and concentrations in 3D fluorescence). It is the reason why we focus on the very important case of nonnegative CP decomposition algorithms. Solutions have already been developed to take into account this nonnegativity constraint (see *e.g.* [3][13][14][15][16][17][18][19]). Their common denominator is that they all rely upon deterministic optimization schemes. The simplest approach consists of iterative alternating minimization schemes (or Alternating Nonnegative Least Squares (ANLS) approaches) where at each iteration the non-negativity constraint is imposed by a projection on the feasible set. This principle is used in the well-known NTF-ALS and NTF-HALS algorithms [3]. The main advantage of this nonnegativity constraint is that the low rank approximation problem becomes well posed [20]. Its counterpart is that its level of difficulty might increase. Unlike other methods, we opt for a direct stochastic algorithm and explain how the nonnegativity constraint is ensured.

Another important aspect is that with our technological capacity to gather, record and store always more

and more information, the volume of available data is continually increasing. Indeed, advanced data mining techniques are required to be able to extract relevant information from this huge amount of data within tolerable elapsed time. In the field of "Big Data", the ability to efficiently process and analyse large data sets has become a key challenge. This is particularly true for tensors as proven by recent articles on this topic (see [21] for an overview). However, very few works have been led on huge nonnegative tensors. A first two-stages solution can be found in [22][23] in which the raw tensor is divided into sub-tensors of smaller sizes simultaneously factorized thanks to distributed computing. This approach is also fast because Kronecker and Khatri-Rao products are avoided and replaced by Hadamard products and multiplication of small matrices. In [24], to speed up the global computational time, the authors suggest a two-steps algorithm with a first stage dedicated to the compression of the original tensor thanks to a Tucker3 decomposition. The stochastic algorithm outlined, here, falls within this "Big Data" framework too. But we suggest a different approach to achieve reduced processing time. Instead of a dimensionality reduction stage like in [24] or [25] (where "random fibers" are used to approximate the unfolding of a high-dimensional tensor in a given mode by a suitable sampling of its columns or rows), we are taking into account the redundancy of information by focusing on a reduced set of randomly chosen equations. The main advantage of such an approach is to offer a higher level of modularity. Two problems can be addressed with exactly the same algorithm i) the non negative CP decomposition (NCP) of tensors, and ii) the NCP decomposition of tensors with possible missing (or unknown, damaged or unreliable) data [26][27][28]. For such problems, the classical "marginalization" approach consists of ignoring unreliable values. With standard approaches, this is achieved at the expense of a modified binary weighted cost function. In our case, the equations corresponding to missing values are simply discarded.

This letter is organized as follows. First, the problem is stated. The objective function as well as the constraints that will be considered are introduced. Then, in Section III, the different aspects of the suggested stochastic algorithm are detailed. In Section IV, its efficiency and gain in computation time are emphasized on synthetic data and it is compared with standard approaches. Finally, a conclusion is drawn.

II. Problem statement

A. The CP model

A tensor can be represented by a L-mode array in a chosen basis. Its order L corresponds to the number of indices of the associated array (*ways* or *modes* [7]). We focus on third-order tensors, say $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$. Each entry of $\mathcal{A}$ is then denoted by a_{ijk}. Such tensors admit the following trilinear

decomposition, also known as the triadic decomposition [29] of $\mathcal{A}$ into a sum of rank-1 tensors *i.e.* for all $(i,j,k) \in \mathcal{I} = [1,\ldots,I] \times [1,\ldots,J] \times [1,\ldots,K] \subset \mathbb{N}^3$:

$$a_{ijk} = \sum_{r=1}^{R} u_{ir} v_{jr} w_{kr}. \tag{1}$$

The three involved matrices $\mathbf{U} = (u_{ir}) = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_R] \in \mathbb{R}^{I \times R}$, $\mathbf{V} = (v_{jr}) = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_R] \in \mathbb{R}^{J \times R}$, $\mathbf{W} = (w_{kr}) = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_R] \in \mathbb{R}^{K \times R}$ are called the *loading matrices*, whose R columns are the so-called *loading factors*. R stands for a large enough integer corresponding to the number of components involved in the sum. The minimum R that can be found such that the above equality remains valid is called the tensor rank and the decomposition is then named the Canonical Polyadic (CP) decomposition of $\mathcal{A}$.

B. CP decomposition of 3-way arrays

A standard way to determine the three loading matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ involved in the CP decomposition consists of minimizing a well chosen objective function $\mathcal{F}$ with respect to these three matrices. The following squared Euclidian distance between the tensor and its approximation *i.e.* the least squares loss function, is frequently used:

$$\mathcal{F}(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{(i,j,k) \in \mathcal{I}} (a_{ijk} - \sum_{r=1}^{R} u_{ir} v_{jr} w_{kr})^2. \tag{2}$$

When performing the CP decomposition, the tensor rank R is assumed to be known. The nonnegativity constraint imposes that $u_{ir} \geq 0$, $v_{jr} \geq 0$ and $w_{kr} \geq 0$ for all i, j, k, r. For huge tensors, I J and K tends to become really high, consequently the calculation of the loss function may require large computational times.

III. A NEW STOCHASTIC ALGORITHM TO SOLVE THE NONNEGATIVE CP DECOMPOSITION PROBLEM

A. A partial objective function

One can see that the system described by Eq. (1) consists of IJK equations with $(I+J+K)R$ unknowns. If the rank R is relatively small compared to $\min(IJ, JK, IK)$, where $\min(.)$ returns the smallest of all the elements within brackets, then the number of unknowns is significantly small compared to the number of equations. Starting from this observation, we suggest to focus on a partial objective function $\mathcal{F}_N$ which is based on the use of N equations randomly chosen among the IJK available equations given by Eq. (1). The value of N should be large enough such that the inequality $(I+J+K)R \leq N \leq IJK$ holds. Then the following notations are introduced to describe $\mathcal{F}_N$. For simplicity, we identify each equation

with the corresponding index $(i,j,k) \in \mathcal{I}$. We denote by $\mathcal{I}_N$ the index set of the N randomly chosen equations:

$$\mathcal{I}_N = \{(i,j,k) \in \mathcal{I} \mid \text{equation } (i,j,k) \text{ is chosen}\}. \tag{3}$$

We also consider the vector $\mathbf{x}$ of size $(I+J+K)R \times 1$:

$$\mathbf{x} = \begin{pmatrix} \mathsf{vec}\{\mathbf{U}\} \\ \mathsf{vec}\{\mathbf{V}\} \\ \mathsf{vec}\{\mathbf{W}\} \end{pmatrix} \tag{4}$$

where operator $\mathsf{vec}\{\cdot\}$ is stacking the columns of a matrix into a vector. Finally, the partial objective function writes:

$$\mathcal{F}_N(\mathbf{x}) = \sum_{(i,j,k)\in\mathcal{I}_N} (a_{ijk} - \sum_{r=1}^{R} u_{ir}v_{jr}w_{kr})^2 \tag{5}$$

Such an approach also constitutes a very simple way to handle the problem of tensors factorization with incomplete or missing data. These values are simply discarded and cannot belong to $\mathcal{I}_N$. Finally, note that when $N = IJK$, $\mathcal{F}_N(\cdot)$ is identical to $\mathcal{F}(\cdot,\cdot,\cdot)$.

B. The general principle of the algorithm

Memetic algorithms are based on an evolution of a population of candidates for the minimization of a function. They are divided into a local search step where each candidate searches around its location, and a selection step where the global population is modified according to some rules like mutation or cloning. Here we use a very simple approach based on only two candidates where only one will survive at the end of each step. At the beginning of the search step, our candidate is located at position $\mathbf{x}$, then local search is done at a position $\hat{\mathbf{x}}$ near $\mathbf{x}$. If $\mathcal{F}_N(\hat{\mathbf{x}}) \leq \mathcal{F}_N(\mathbf{x})$ then the new position is $\hat{\mathbf{x}}$, if not the candidate stays at position $\mathbf{x}$. In terms of memetic algorithms, $\mathbf{x}$ is cloned, its clone goes to position $\hat{\mathbf{x}}$ and then there is a binary tournament between $\mathbf{x}$ and $\hat{\mathbf{x}}$ to determine who will survive. This technic requires only one evaluation of the objective function at each step instead of matrix inversion or computation of derivatives for usual deterministic methods. The objective function does not need to be regular like for gradient methods which makes the algorithm highly flexible. Its adaptation to other (possibly more complicated) problems only involves a change of the cost function (no derivative).

C. Stochastic local search

There are obviously many ways to search around position $\mathbf{x}$ to create a new candidate $\hat{\mathbf{x}}$. We choose here to modify at each step only one component of $\mathbf{x}$ to build our candidate $\hat{\mathbf{x}}$. This particular choice is

motivated by a low cost computation of $\mathcal{F}_N(\hat{\mathbf{x}})$ thanks to the value of $\mathcal{F}_N(\mathbf{x})$ as described in the next section. This component is chosen uniformly at random among the $(I+J+K)R$ elements of $\mathbf{x}$ that is in the interval $[\![1,(I+J+K)R]\!]$. It is modified thanks to a random variable Y. Without loss of generality, if we suppose that the element $u_{i_0r_0}$ of $\mathbf{x}$ is chosen, then $\hat{\mathbf{x}}$ is defined as follows

$$\hat{u}_{ir} = \begin{cases} |u_{i_0r_0} + Y| & \text{if } (i,r) = (i_0,r_0), \\ u_{ir} & \text{otherwise,} \end{cases} \tag{6}$$

$$\hat{v}_{jr} = v_{jr}, \tag{7}$$

$$\hat{w}_{kr} = w_{kr}. \tag{8}$$

The non-negativity of $\hat{\mathbf{x}}$ is simply ensured thanks to (6) since $|\cdot|$ stands for the absolute value. The crucial question is the choice of the law of Y and of its parameters. As we want to search around the current position, the law of Y should be symmetric. We have chosen here to use a uniform law in an interval $[-s_p, s_p]$ which will depend on the iteration step denoted by p but not on a particular component of $\hat{\mathbf{x}}$. To achieve an acceptable rate of convergence for our method the bound s_p should be a decreasing function of p. For usual stochastic algorithms like the stochastic gradient with decaying step-size, it is proven that if the rate of decay is chosen a priori and verifies $\sum_{p=1}^{\infty} s_p = \infty$ and $\sum_{p=1}^{\infty} s_p^2 < \infty$ the convergence to a local minimum is granted [30]. A frequently used step-size is $s_p = 1/p$. Here our approach is closer to deterministic methods as our step depends on $\mathcal{F}_N$ and gets smaller when $\mathcal{F}_N$ gets smaller. The computation of our step relies on two heuristics. First, the quantity $\sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}$ indicates the mean error that we have on each term $|a_{ijk} - \sum_{r=1}^{R} u_{ir}v_{jr}w_{kr}|$ belonging to $\mathcal{F}_N$. Then, we can try to improve our search using the following scaling. If we assume that all components of $\mathbf{x}$ are equal *i.e.* $u_{ir} = v_{jr} = w_{kr} = \tau$, then the solution of the minimization problem is $\tau = \sqrt[3]{\frac{\sum_{(i,j,k)\in\mathcal{I}} a_{ijk}}{IJKR}}$. Indeed in this case, thanks to (1) we have $\sum_{(i,j,k)\in\mathcal{I}} a_{ijk} = \sum_{(i,j,k)\in\mathcal{I}} \sum_{r=1}^{R} \tau^3 = IJKR\tau^3$ (note that we also use the quantity τ for our initialization). To modify for instance the law of $u_{i_0r_0}$ in $a_{i_0jk} - \sum_{r=1}^{R} u_{i_0r}v_{jr}w_{kr}$ we choose a random variable Y such that $Yv_{jr_0}w_{kr_0}$ is a uniform law in $(-\sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}, \sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}})$. Finally, thanks to the second heuristic the law of Y is uniform in $(-\sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}/\tau^2, \sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}/\tau^2)$.

D. A low-cost computation of the objective function

The objective function $\mathcal{F}_N(\hat{\mathbf{x}})$ can be computed in an inexpensive way thanks to $\mathcal{G}_N(\hat{\mathbf{x}},\mathbf{x}) = \mathcal{F}_N(\hat{\mathbf{x}}) - \mathcal{F}_N(\mathbf{x})$. Regarding (6), $\hat{\mathbf{x}}$ and $\mathbf{x}$ have only one different component and $\mathcal{G}_N(\hat{\mathbf{x}},\mathbf{x})$ can be rewritten as the

difference between the sum of all quadratic terms in (5) containing $u_{i_0r_0}$. Hence $\mathcal{G}_N(\hat{\mathbf{x}}, \mathbf{x})$ is equal to

$$\sum_{(i,j,k)\in\mathcal{I}_N} \left[(a_{ijk} - \sum_{r=1}^{R} \hat{u}_{ir}\hat{v}_{jr}\hat{w}_{kr})^2 - (a_{ijk} - \sum_{r=1}^{R} u_{ir}v_{jr}w_{kr})^2 \right] =$$
$$\sum_{(j,k)\in\mathcal{I}_N^{i_0}} \left[(a_{i_0jk} - \sum_{r=1}^{R} \hat{u}_{i_0r}\hat{v}_{jr}\hat{w}_{kr})^2 - (a_{i_0jk} - \sum_{r=1}^{R} u_{i_0r}v_{jr}w_{kr})^2 \right] \quad (9)$$

where $\mathcal{I}_N^{i_0} = \{(j,k) \mid (i_0,j,k) \in \mathcal{I}_N\}$. Factorizing each term in (9) and using $\hat{u}_{i_0r_0} - u_{i_0r_0} = \Delta u$ finally leads to the following more compact form

$$\mathcal{G}_N(\hat{\mathbf{x}}, \mathbf{x}) = \sum_{(j,k)\in\mathcal{I}_N^{i_0}} \Delta u v_{jr_0} w_{kr_0} \alpha_{i_0jkr_0}, \quad (10)$$

where $\alpha_{i_0jkr_0} = 2(a_{i_0jk} - \sum_{r=1}^{R} \hat{u}_{i_0r}\hat{v}_{jr}\hat{w}_{kr}) - \Delta u v_{jr_0} w_{kr_0}$.

While each computation of the objective function $\mathcal{F}_N(\hat{\mathbf{x}})$ based on its definition (5) requires $N(3R+1)$ arithmetic operations, computing $\mathcal{G}_N(\hat{\mathbf{x}}, \mathbf{x})$ requires only $(3R+6)\mathsf{card}(\mathcal{I}_N^{i_0})$ operations, where $\mathsf{card}(\cdot)$ stands for the cardinal number of a set. Since $\mathcal{I}_N = \bigcup_{i=1}^{I} \mathcal{I}_N^i$, we expect $\mathsf{card}(\mathcal{I}_N^{i_0})$ to be close to $\frac{N}{I}$ in average. Then, if an entry of $\mathbf{U}$ (resp. $\mathbf{V}$ or $\mathbf{W}$) is chosen, the complexity of each computation of $\mathcal{F}_N(\hat{\mathbf{x}})$ is divided by a factor I (resp. J or K). Finally, to avoid roundoff errors, we propose to recompute periodically the objective function after a large number of iterations M_0 ($M_0 \ll M$ if M is the total number of iterations).

E. Algorithm

Data: 3-way tensor $\mathcal{A}$, R tensor rank, ϵ stopping criterion, M max. number of iterations, M_0 for reevaluation, N number of chosen equations

Result: Estimation of $\mathbf{x} = \text{vec}\{\mathbf{X}\}$, $\mathbf{X} = (\mathbf{U}, \mathbf{V}, \mathbf{W})^T$ with $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ loading matrices involved in CP decomposition of $\mathcal{A}$

Initialize: $\mathbf{x} \geq 0$, calculate $\mathcal{F}_N(\mathbf{x}), \tau$, $p = 1$;

while *($\mathcal{F}_N(\mathbf{x}) \geq \epsilon$) and ($p \leq M$)* **do**

- $\hat{\mathbf{x}} = \mathbf{x}$
- Choose index k uniformly at random in $[\![1, (I+J+K)R]\!]$
- Draw Y uniformly in $(-\sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}/\tau^2, \sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}/\tau^2)$
- Update entry $\widehat{x}_{k_{p+1}}$ of $\hat{\mathbf{x}}$ according to $\widehat{x}_{k_{p+1}} = |x_{k_p} + Y|$;
- **if** $p \mod M_0 \neq 0$ **then**
  - $\mathcal{F}_N(\hat{\mathbf{x}}) = \mathcal{F}_N(\mathbf{x}) + \mathcal{G}_N(\hat{\mathbf{x}}, \mathbf{x})$
- **else**
  - Calculate $\mathcal{F}_N(\hat{\mathbf{x}})$ using definition (5)
- **end**
- **if** $\mathcal{F}_N(\hat{\mathbf{x}}) \leq \mathcal{F}_N(\mathbf{x})$ **then**
  - $\mathbf{x} = \hat{\mathbf{x}}$
- **end**
- $p = p + 1$

end

IV. Computer simulations on simulated data

In this section, we illustrate the behavior of the suggested algorithm on data that have been numerically simulated. Then, we compare our results with classical algorithms of the literature[31][32][33]. We consider the case of loading matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ composed of $R = 5$ columns and respectively $I = J = K = 100$ rows. Their elements are drawn from the standard uniform distribution $\mathcal{U}(0, 1)$. Thus, a $100 \times 100 \times 100$ nonnegative tensor $\mathcal{A}$ is generated. All simulations are performed using the same initialization for $\mathbf{x}$ and $\hat{\mathbf{x}}$ whose elements are generated according to a uniform distribution $\mathcal{U}(0, 2\tau)$. The step-size Y follows a uniform law $\mathcal{U}(-\sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}/\tau^2, \sqrt{\frac{\mathcal{F}_N(\mathbf{x})}{N}}/\tau^2)$. The classical relative reconstruction error $E_1 = \frac{\|\mathcal{A}-\widetilde{\mathcal{A}}\|_F}{\|\mathcal{A}\|_F}$, or $E_{1\text{dB}} = 10\log_{10}(E_1)$ (where $\|.\|_F$ is the Frobenius norm) is used to compare the suggested approach with classical algorithms of the literature (Tab. 1). In order to be able to better assess the behavior of our approach in Fig. 1 & 2, another error index is used:

$$E_{2\text{dB}} = \min_{\sigma \in S_N} 10\log_{10}\left(\frac{\|\mathbf{x}_\sigma - \widetilde{\mathbf{x}}\|_1}{\|\widetilde{\mathbf{x}}\|_1}\right) \tag{11}$$

where $\|.\|_1$ stands for the l_1-norm, vector $\widetilde{\mathbf{x}}$ defined as (4) is a solution of (1) and S_N is the set of all permutations σ of $(1, \ldots, N)$, thus $\mathbf{x}_\sigma \overset{\text{def}}{=} (\mathbf{x}_{\sigma(1)}, \ldots, \mathbf{x}_{\sigma(N)})$. Moreover a normalization is applied to any vector $\mathbf{x}$ since we consider: $\mathbf{u}_r = \frac{\mathbf{u}_r}{\|\mathbf{u}_r\|_1}$, $\mathbf{v}_r = \frac{\mathbf{v}_r}{\|\mathbf{v}_r\|_1}$, and $\mathbf{w}_r = \mathbf{w}_r\|\mathbf{u}_r\|_1\|\mathbf{v}_r\|_1$, for $r =$

$1, \ldots, N$. It is greedier but guarantees the accuracy of the found solution. The algorithm is then stopped when the number of iterations M is greater than 10^7 or when the minimum value of the two objective functions $\min(\mathcal{F}_N(\mathbf{x}), \mathcal{F}_N(\hat{\mathbf{x}}))$ is smaller than $\epsilon = 10^{-16}$. The objective functions are reevaluated at every $M_0 = 2 \times 10^4$ iterations. First, our aim is to evaluate the impact on the obtained performance of the number N of equations involved in the considered partial cost function $\mathcal{F}_N$. The percentage of chosen equations is equal to $\frac{100N}{IJK}$. We consider values of N corresponding to the following percentages: $(0.15\%, 0.22\%, 0.3\%, 0.45\%, 0.6\%, 0.75\%, 1.2\%, 1.5\%, 7.5\%,\ 100\%)$. Results are displayed on Fig. 1. Moreover, for each value of N, 7 different random subsystems of N equations involved in $\mathcal{F}_N$ have been considered. The results have been averaged over those 7 different trials. The best and worst performance among those 7 runs are also plotted. This chart emphasizes the great redundancy of information of the considered problem. In fact, on this example, we were able to solve the CP decomposition problem considering only 0.45% of equations without too significant performance degradation (-83db were still reached). As illustrated on Fig. 2, the computation time depends on the percentage of considered equations. By diminishing this percentage, the computation speed can be improved. However, when less and less equations are considered, the computation speed increases again, and finally when too few equations are considered, the CP decomposition problem cannot be solved anymore. In this example, 0.6% offers the best compromise between performance and computation speed. Finally, in Tab. I, the proposed algorithm, NTF-STO (with 3% of equations), is compared with classical algorithms of the literature (NTF-ALS, fast NTF-HALS, Bro's $N-way$). It can be observed that the proposed algorithm becomes more competitive when the tensor dimensions increase, but not for too high ranks. Yet, it remains more general-purpose.

I	R	NTF-ALS	fast NTF-HALS	$N-way$	NTF-STO
100	5	15s	2.76s	11s	216s
200	5	113s	8.74s	52s	383s
400	5	957s	46s	388s	1283s
500	5	1746s	$99s$	662s	1556s
100	10	32s	15s	20s	2508 s

TABLE I: Running time for the NTF-ALS, fast NTF-HALS, Bro's $N-way$ and our stochastic algorithm for different sizes $I = J = K$ of tensors, and a rank $R = 5$ (or 10). Stopping criterium: $E_1 < 1e-8$ ($7e-7$ for fast-HALS which cannot reach the same level of performance).

V. CONCLUSION

We have presented a new approach based on a stochastic algorithm to handle the nonnegative CP decomposition of large three-way tensors. It can be seen as a very special case of memetic algorithms.

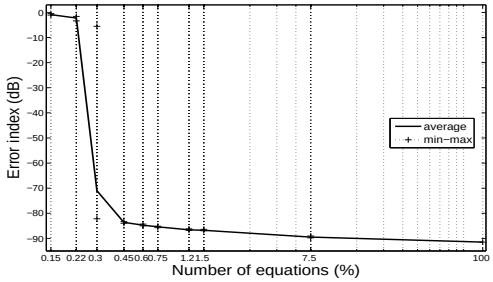

Fig. 1: Performance versus $\%$ of considered equations.

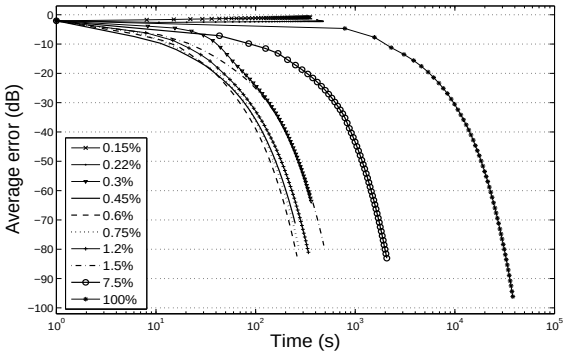

Fig. 2: Performance versus computation speed for different percentages of considered equations.

Computer simulations led on synthetic data have been provided to emphasize the efficiency of this approach both in terms of performance and computation time (but not for too high tensor ranks). This version is not yet fully optimized and further works will consist in improvements in the step-size choice and smart pre-stocking of certain quantities involved in the cost function calculation to reduce the computation time even further. It could also be generalized to the factorization of nonnegative L-way arrays with $L > 3$ or to tackle more complicated factorization problems.

REFERENCES

[1] E. Acar and B. Yener, "Unsupervised multiway data analysis: a literature survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 1, pp. 6–20, Jan. 2009.

[2] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *Siam Review*, vol. 51, no. 3, pp. 455–500, Sept. 2009.

[3] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Non negative matrix and tensor factorizations: Application to exploratory multi-way data analysis and blind separation*, Wiley, 2009.

[4] P. Comon, "Tensor: a brief introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 44–53, May 2014.

[5] L. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, pp. 279–311, 1966.

[6] E. Sanchez and B.R. Kowalski, "Tensorial resolution: a direct trilinear decomposition," *J. Chemometrics*, vol. 4, pp. 29–45, 1990.

[7] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis with applications in the chemical sciences*, Wiley, 2004.

[8] P. Moscato, *Memetic algorithms: A short introduction,*, vol. New Ideas in Optimization, pp. 219–234, McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999.

[9] P. Moscato and C. Cotta, *A gentle introduction to memetic algorithms*, vol. Handbook of Metaheuristics, pp. 105–144, Kluwer Academic Publishers, Boston, MA, 2003.

[10] C. A. Stedmon, S. Markager, and R. Bro, "Tracing dissolved organic matter in aquatic environments using a new approach to fluorescence spectroscopy," *Marine Chemistry*, vol. 82, pp. 239–254, 2003.

[11] J.-P. Royer, N. Thirion-Moreau, P. Comon, R. Redon, and S. Mounier, "A regularized nonnegative canonical polyadic decomposition algorithm with preprocessing for 3d fluorescence spectroscopy," *Journal of Chemometrics*, vol. 29, pp. 253–265, March 2015.

[12] Q. Zhang, H. Wang, R. Plemmons, and P. Pauca, "Tensors methods for hyperspectral data processing: a space object identification study," *Journal of Optical Society of America A*, Dec. 2008.

[13] R. Bro and S. De Jong, "A fast non-negativity constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.

[14] P. Paatero, "A weighted non-negative least squares algorithm for three-way Parafac factor analysis," *Chemometrics Intell. Lab. Systems*, vol. 38, no. 2, pp. 223–242, 1997.

[15] A. H. Phan, P. Tichavskỳ, and A. Cichocki, "Low complexity damped gauss-newton algorithms for candecomp/parafac," *SIAM. J. Matrix Anal. & Appl.*, vol. 34, no. 1, pp. 126–147, 2013.

[16] J.-P. Royer, N. Thirion-Moreau, and P. Comon, "Computing the polyadic decomposition of nonnegative third order tensors," *Eurasip Signal Processing*, vol. 91, no. 9, pp. 2159–2171, Sept. 2011.

[17] J.-P. Royer, P. Comon, and N. Thirion-Moreau, "Computing the nonnegative 3-way tensor factorization using tikhonov regularization," in *International Conference on Acoustic Speech and Signal Processing (ICASSP'2011)*, Prague, Czech Republic, May 2011, pp. 2732–2735.

[18] D. Bunker, L. Han, and S. Zhang, "A proximal anls algorithm for nonnegative tensor factorization with a periodic enhanced line search," *Application of Mathematics*, vol. 58, no. 5, pp. 493–509, Oct. 2013.

[19] J. Kim and H. Park, "Fast nonnegative tensor factorization with an active-set-like method," *High-Performance Scientific Computing: Algorithms and Applications, Springer*, pp. 311–326, 2012.

[20] L. H. Lim and P. Comon, "Nonnegative approximations of nonnegative tensors," *Jour. Chemometrics*, vol. 23, pp. 432–441, Aug. 2009.

[21] N. Sidiropoulos, E. E. Papalexakis, and C. Faloutsos, "Parallel randomly compressed cubes," *IEEE Signal Processing Magazine, Special Issue on Big Data*, vol. 31, no. 5, pp. 57–70, 2014.

[22] A. H. Phan and A. Cichocki, "Fast nonnegative tensor factorization for very large-scale problems using two-stage procedure," in *3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2009.

[23] A. H. Phan and A. Cichocki, "Parafac algorithms for large scale problems," *Neurocomputing*, vol. 74, pp. 1970–1984, June 2011.

[24] J. Cohen, R. Cabral Farias, and P. Comon, "Fast decomposition of large nonnegative tensors," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 862–866, July 2015, Online since Nov. 2014.

[25] C. F. Caiafa and A. Cichocki, "Generalizing the column-row matrix decomposition to multi-way arrays," *Linear Algebra and its Applications*, vol. 433, no. 3, pp. 557–573, 2010.

[26] E. Acar, T. G. Kolda, D. M. Dunlavy, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chem. Intel. Lab. Syst.*, vol. 106, pp. 41–56, Jan. 2011.

[27] G. Tomasi and R. Bro, "Parafac and missing values," *Chemometrics Intell. Lab. Systems*, 2004.

[28] J.-P. Royer, N. Thirion-Moreau, and P. Comon, "Nonnegative 3-way tensor factorization taking into account possible missing data," in *European Signal Processing Conference (EUSIPCO'2012)*, Bucharest, Romania, August 2012.

[29] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. and Phys.*, vol. 6, pp. 165–189, 1927.

[30] L. Ljung, "Analysis of stochastic gradient algorithms for linear regression problems," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 151–160, March 1984.

[31] A. H. Phan, P. Tichavskỳ, and A. Cichocki, "Tensorbox: a matlab package for tensor decomposition," available online at http://www.bsp.brain.riken.jp/ phan/tensorbox.php, 2013.

[32] R. Bro and C. A. Andersson, "N-way toolbox for matlab," http://www.models.life.ku.dk/nwaytoolbox, 2014.

[33] A. H. Phan, P. Tichavskỳ, and A. Cichocki, "Fast alternating ls algorithms for high order candecomp/parafac tensor factorizations," *IEEE Trans. on Sig. Proc.*, vol. 61, no. 19, pp. 4834–4846, June 2013.